The phrase decouple deployments from releases refers to the practice of separating the technical process of deploying code to a production environment from the decision of when to make that code publicly available to accessible. IN traditional software delivery models, the deployment and the release are often tightly coupled meaning that when you deploy your code to production, it becomes immediately available to users. By decoupling these two activities, it allows us to deploy changes to production without necessarily releasing them to the users.

Some reasons why organizations might decouple the deployment and releases are that it can lead to faster deployment cycles. You can deploy frequently without worrying about the time of release. It would also be safe for you to test new features in production with limited exposure to internal users before it affects the users. This allows us to test production with real world data before releasing it to the users, which makes it easier to identify and fix issues in a controlled way.

As great as it sounds, there are some challenges that might lead an organization astray from decoupling. It can lead to increased complexity, managing things like feature flags and toggles. If everything isn’t managed well, it could be hard to manage it all. It could also lead to a lack of visibility having hidden flags here and hidden flags there and again without the proper documentation of each hidden feature it would be difficult for other employees and shareholders to see what feature what is. To which it could also lead to more bugs. I know we have all experienced the frustrations of that one setting that randomly got turned off, or that one setting that is interfering with another. That same concept would apply here.

Decoupling deployments from releases is not only possible but also beneficial for many modern software development practices. It offers flexibility, reduces risk, and enables faster, safer delivery. However, it requires good practice in managing complexity and requires teams to be disciplined about how they use feature flags and deployment strategies. For organizations that are not ready or able to handle the added complexity, sticking with a more traditional, tightly coupled deployment and release process might be a better fit.

https://posthog.com/product-engineers/decouple-deployment-from-release